Final Exam:

PHY 410 : Do problems 1-3.
PHY 505: Do all 4 problems.

Store all of your codes in a folder called “Final” at the top level of your directory. Please accept the assignment here: https://classroom.github.com/a/PfA9xUjJ

Name your codes “Problem1.ipynb”, “Problem2.ipynb”, “Problem3.ipynb” and “Problem4.ipynb”.

For Problems 1 and 2, consider the effective potential of a co-rotational frame in a 2-d 3-body system as we discussed in class:

$$V(x', y') = -\frac{1-\alpha}{\sqrt{(x'-\alpha)^2 + y'^2}} - \frac{\alpha}{\sqrt{(x'+1-\alpha)^2 + y'^2}} - \frac{x'^2 + y'^2}{2}$$

where x' and y' are the horizontal and vertical dimensions in the co-rotating reference frame.

For the remainder of the discussion, we will write these as x and y, but do not get confused, these are still the rotating frame. Use $\alpha = 0.25$ for all problems.

Problem 1 (25 Points) : Finding Lagrange Points

Part a (10 points): Create a separate `python` file called `V_coriolis.py` with a class called `V_coriolis`, with at least two methods `f` (to implement the function above) and `df` (to implement the gradient of the function above). The function `f` should return a single floating point number, and the function `df` should return the x and y values of the gradient in a list.

Part b (25 points):
Plot the 2-d function as a contour plot.
ON THE SAME FIGURE, also plot the gradient using the quiver function.

Use the following parameters:
* Plot a square with $-2 < x, y < 2$ with 101 grid spacing points in each direction.
* Use 200 levels on the contour plot.
* Use a "stride" of 5 to only plot 1 out of 5 of the vectors in the vector field in the `quiver` plot.

Problem 2 (25 Points): Trajectories

Part a (5 points): Starting with the notebook “ODEs/planetary.ipynb”, copy the class “Rcp3Body” to a file called “Rcp3Body.py” in your “Final” directory. There is a chaotic dynamical system if you use the initial values in the notebook titled “params_rcp3body_chaotic”:

```
params_rcp3body_chaotic = {
    'tau':0.01,
    'accuracy':1e-6,
    'a':0.25,
    'method':'RK45',
    's':np.array([-0.35, 0.0, 0.0, 0.5]),
    'tmax':10.
}
```

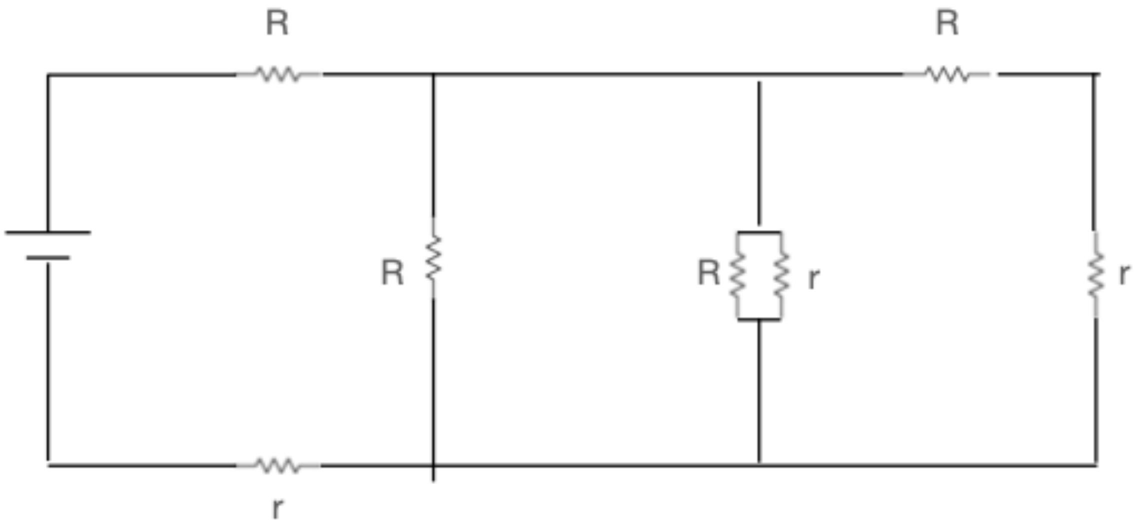
* Run that simulation and plot x vs y as a scatter plot, labeling all of the Lagrange points as is done in the example code. **Extend the time to t=100**.
* Plot the following on the same plot, label them, and draw a legend:
  * x vs time
  * x vs time
  * vx vs time
  * vy vs time

Part b (10 points): Take the FFT of all four time series from part a (x,y,vx,vy) and plot the power series (not the absolute value!) on a log scale. Truncate your time series at n=2048 (do not perform padding). You may also use numpy functions.

Part c (10 points): Zero out the components of the FFT spectra above a spectral index of 80. Take the inverse FFT and plot the x vs t time series with and without this smoothing procedure on the same plot, labeling each. Also perform the same with y and plot x versus y on the same plot with and without smoothing (with the same number of points n=2048), labeling each.

Problem 3 (25 Points) : Resistor circuit

Consider the following resistor circuit with R = 1 Ohm and r varying between 0 and 2 Ohms.



Part a (10 points):
Solve this analytically and plot the equivalent resistance for r between 0 and 2.

Part b (15 points):
Solve for the equivalent resistance using a matrix inversion and plot the equivalent resistance for r between 0 and 2.

Problem 4 (25 Points): Particle in harmonic potential

Part a (10 points): Starting with the notebook "PDEs/wavepacket.ipynb", modify the simulation to have a harmonic oscillator potential at the center of the box with energy $V_0 = 10$, and the width should be half the length of the box. That is, the formula for the potential should be

$V(x) = V_0 \left((x - x_0)/w\right)^2$

where $V_0 = 10$ is the height of the potential, x_0 is the middle of the box, and w should be half the length of the box

Part b (15 points): Animate the potential as shown in the "wavepacket" notebook in "PDEs". I will be executing the notebook so ensure that it works out of the box when you upload it all to GitHub!